(12) UK Patent Application (19) GB (11) 2 365 243 (13) A

(43) Date of A Publication 13.02.2002

(21) Application No 0018492.9

(22) Date of Filing 27.07.2000

(51) INT CL⁷
G06T 7/00

(52) UK CL (Edition T)
H4F FDX F15
H4T TBBA T131

(56) Documents Cited
EP 0898245 A1

(58) Field of Search
UK CL (Edition S) H4F FCCC FDX
INT CL⁷ G06T 7/00 7/60 9/20 17/00
Online: WPI, EPODOC, JAPIO, INSPEC

(71) Applicant(s)
Canon Kabushiki Kaisha
(Incorporated in Japan)
30-2 3-chome Shimomaruko, Ohta-ku, Tokyo, Japan

(72) Inventor(s)
Alexander Ralph Lyons
Charles Stephen Wiles
Simon Michael Rowe
Jane Haslam
Richard Ian Taylor

(74) Agent and/or Address for Service
Beresford & Co
2-5 Warwick Court, High Holborn, LONDON,
WC1R 5DH, United Kingdom

(54) Abstract Title
Creating a 3D model from a series of images

(57) A 3D model is created from a series of images taken at different camera positions. A sequence of images is processed to obtain co-ordinates of matching features to define a model in 3D space and to obtain camera solutions for the position and orientation of virtual cameras in 3D space defining views of the model. A new image is then added to the sequence corresponding to a new virtual camera. The new image is processed to identify points which match with points in previous images and to find a solution representing the position and orientation of the new virtual camera. The new solution is determined using 2D image co-ordinates and 3D model point co-ordinates.

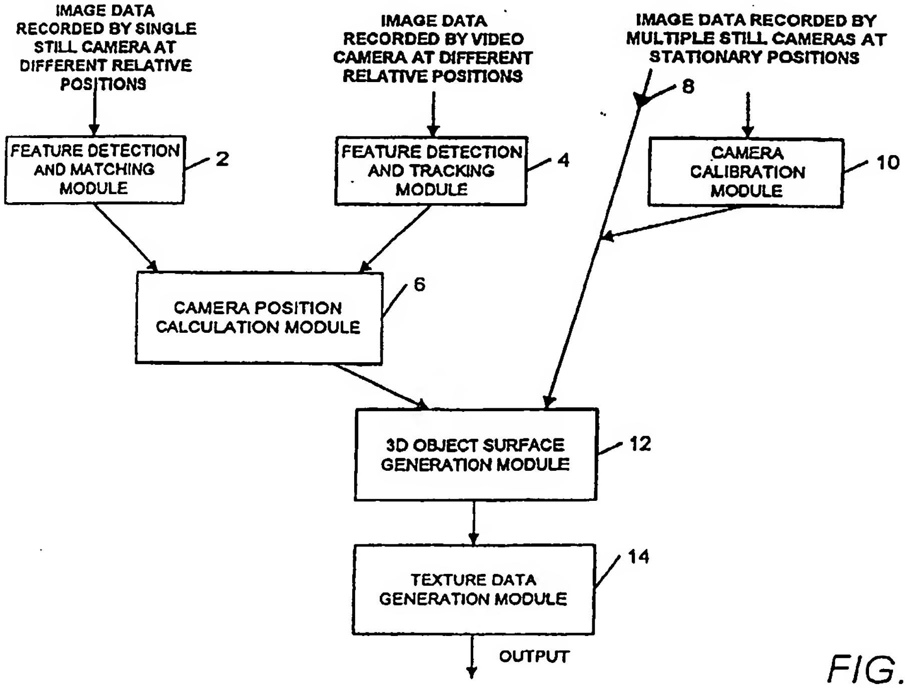
IMAGE DATA RECORDED BY SINGLE STILL CAMERA AT DIFFERENT RELATIVE POSITIONS
IMAGE DATA RECORDED BY VIDEO CAMERA AT DIFFERENT RELATIVE POSITIONS
IMAGE DATA RECORDED BY MULTIPLE STILL CAMERAS AT STATIONARY POSITIONS
8
FEATURE DETECTION AND MATCHING MODULE
2
FEATURE DETECTION AND TRACKING MODULE
4
CAMERA CALIBRATION MODULE
10
CAMERA POSITION CALCULATION MODULE
6
3D OBJECT SURFACE GENERATION MODULE
12
TEXTURE DATA GENERATION MODULE
14
OUTPUT

FIG. 1

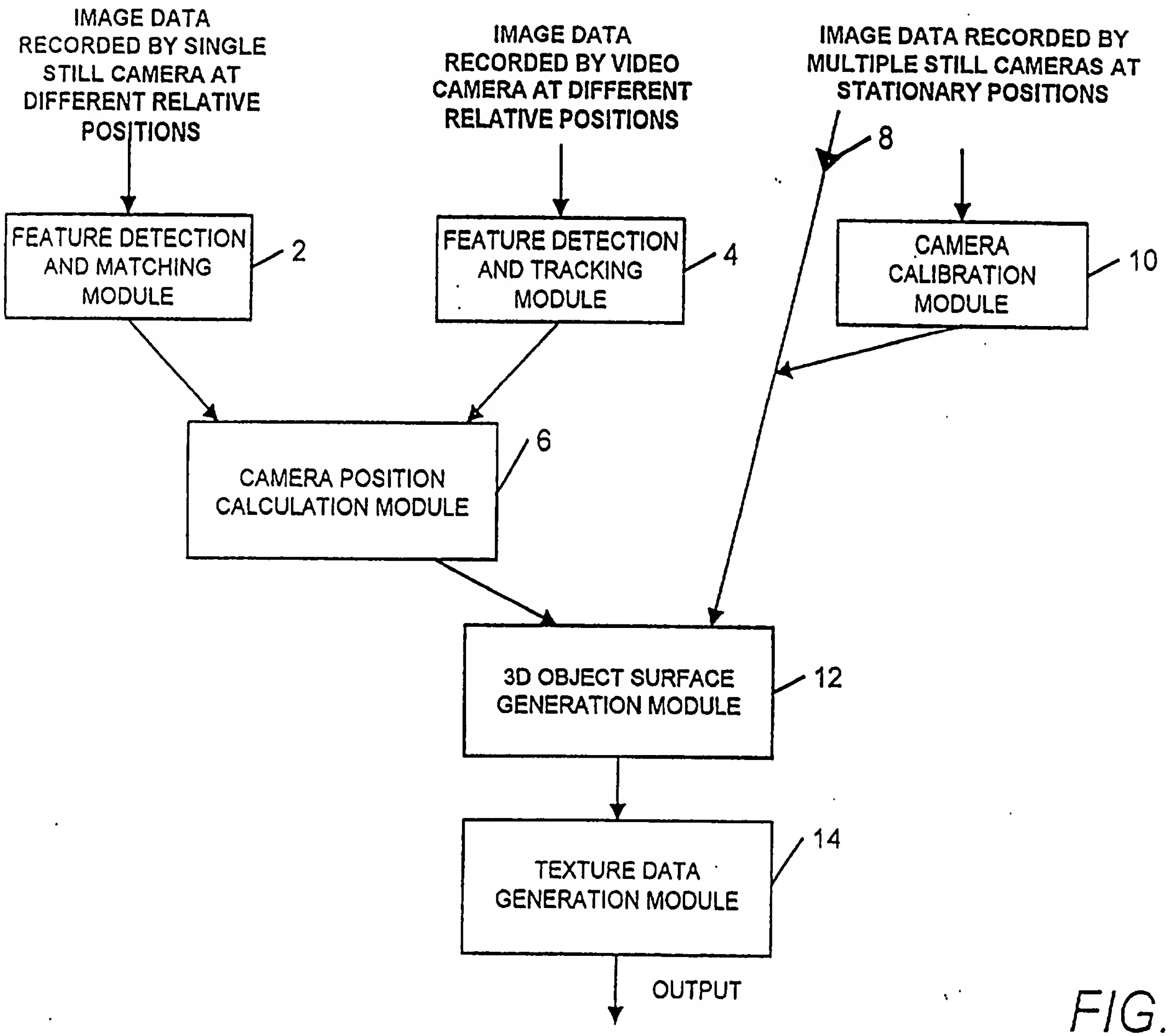
IMAGE DATA RECORDED BY SINGLE STILL CAMERA AT DIFFERENT RELATIVE POSITIONS
IMAGE DATA RECORDED BY VIDEO CAMERA AT DIFFERENT RELATIVE POSITIONS
IMAGE DATA RECORDED BY MULTIPLE STILL CAMERAS AT STATIONARY POSITIONS
8
FEATURE DETECTION AND MATCHING MODULE
2
FEATURE DETECTION AND TRACKING MODULE
4
CAMERA CALIBRATION MODULE
10
CAMERA POSITION CALCULATION MODULE
6
3D OBJECT SURFACE GENERATION MODULE
12
TEXTURE DATA GENERATION MODULE
14
OUTPUT

FIG. 1

Fig 2A

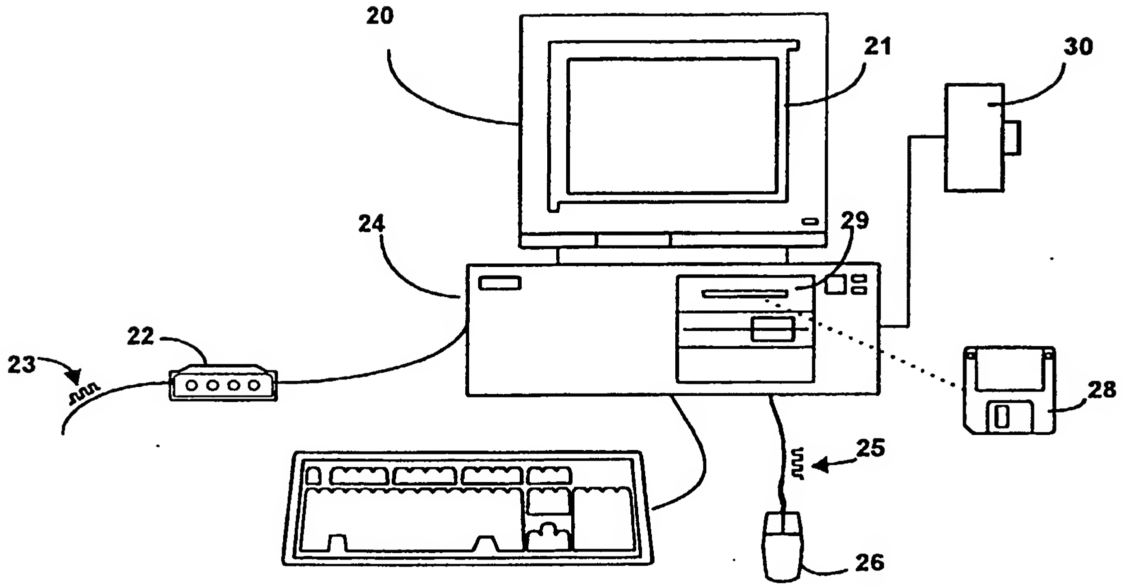
20
21
30
24
29
22
23
28
25
26

Fig 2B

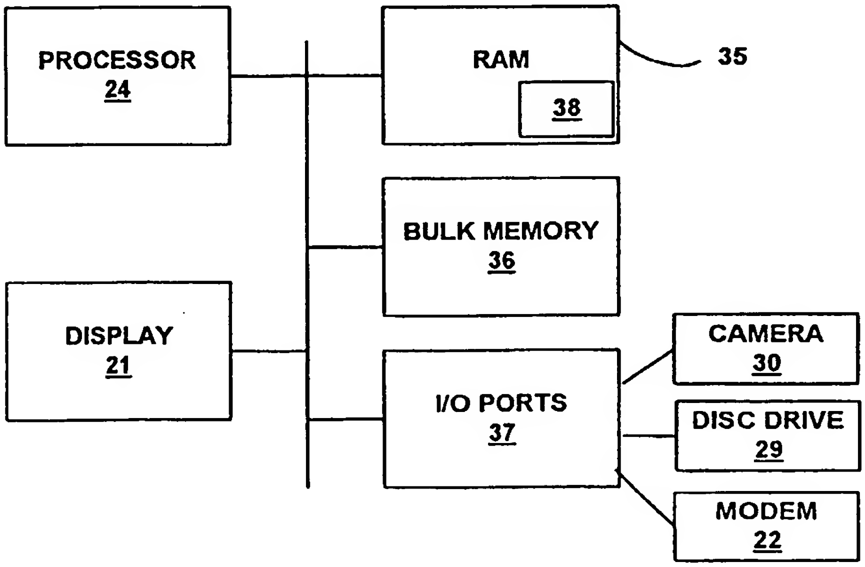
PROCESSOR
24
RAM
38
35
BULK MEMORY
36
DISPLAY
21
CAMERA
30
I/O PORTS
37
DISC DRIVE
29
MODEM
22

Fig 3A

ACTUAL CAMERA POSITIONS RELATIVE TO THE OBJECT

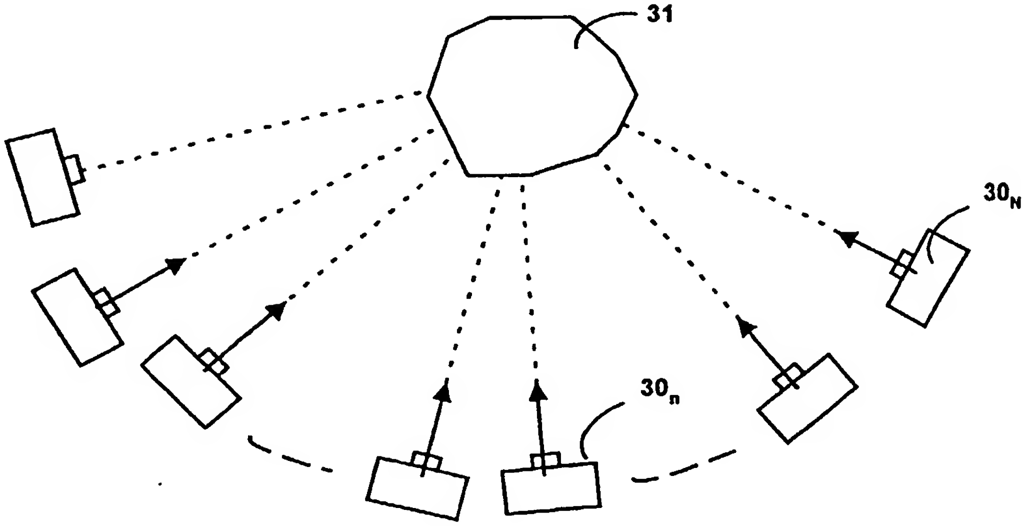

Fig 3B

VIRTUAL CAMERA POSITIONS RELATIVE TO THE MODEL

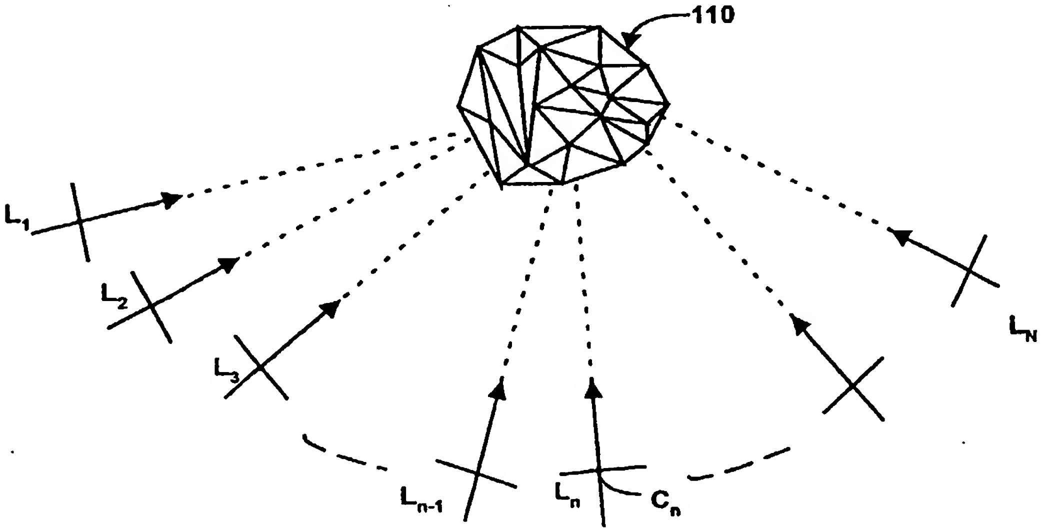

Fig 4

PERFORMING MATCHING BETWEEN FEATURES OF NEW IMAGE AND PRECEDING IMAGE

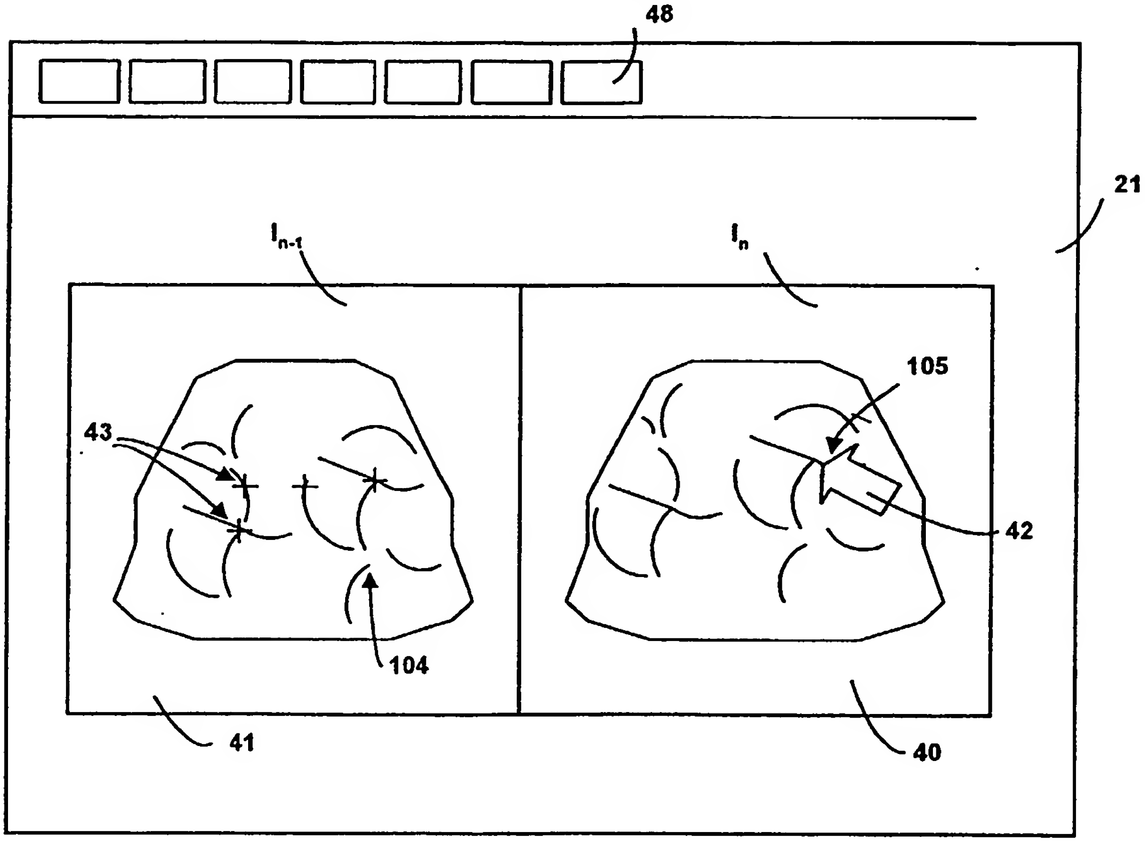

Fig 5

MAPPING EXISTING MODEL POINTS INTO THE VIRTUAL IMAGE PLANE AT THE VIRTUAL CAMERA POSITION OF THE NEW IMAGE USING THE PROVISIONAL CAMERA SOLUTION (STEP 67)

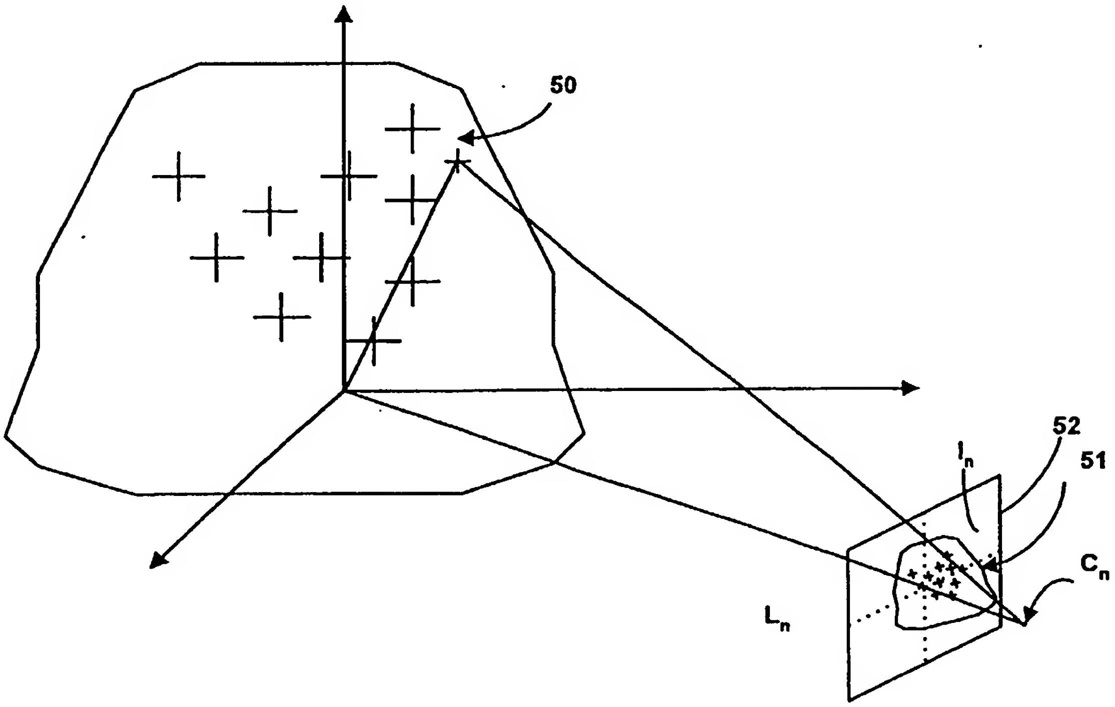

FIG. 6A

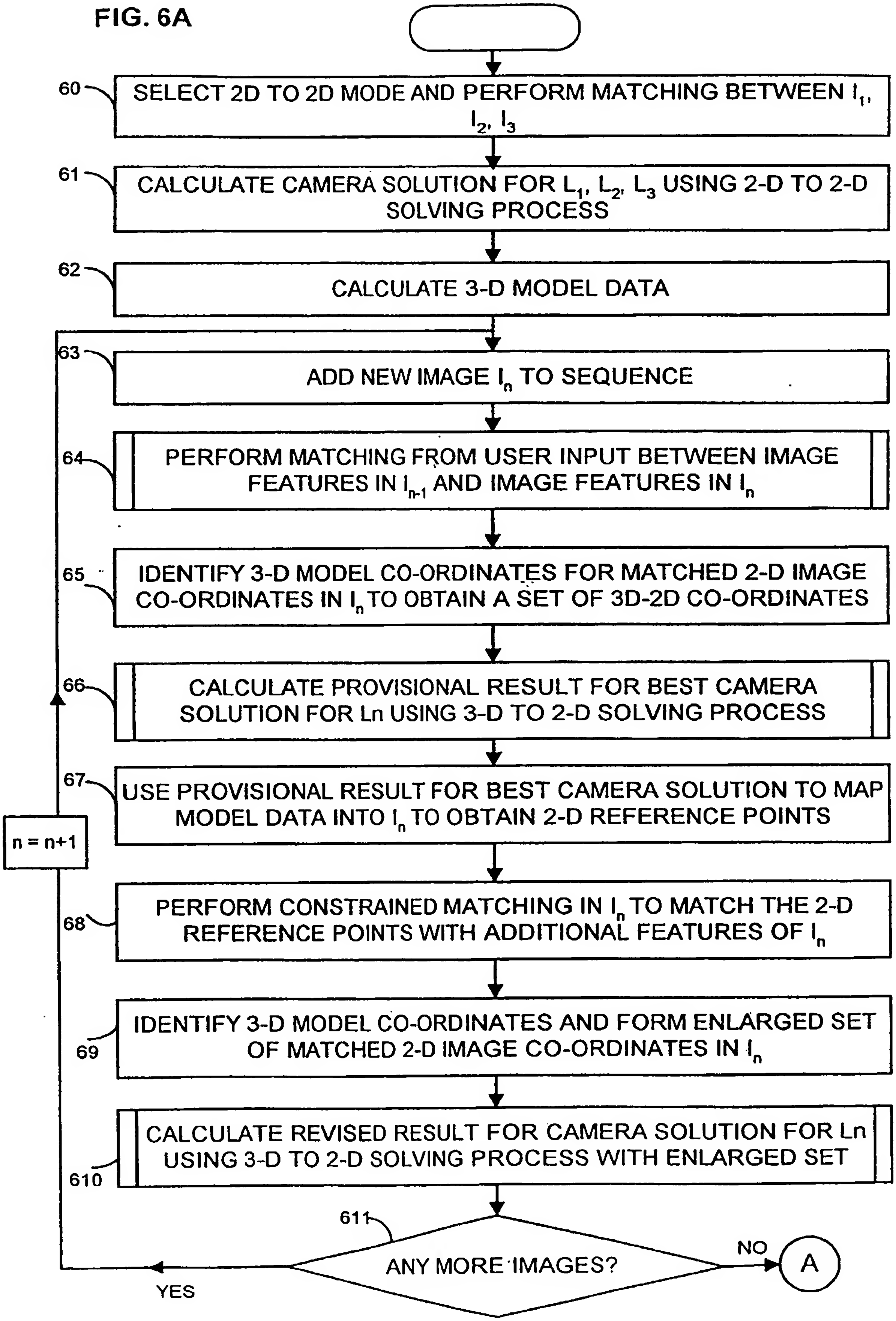
60
SELECT 2D TO 2D MODE AND PERFORM MATCHING BETWEEN I_1, I_2, I_3
61
CALCULATE CAMERA SOLUTION FOR L_1, L_2, L_3 USING 2-D TO 2-D SOLVING PROCESS
62
CALCULATE 3-D MODEL DATA
63
ADD NEW IMAGE I_n TO SEQUENCE
64
PERFORM MATCHING FROM USER INPUT BETWEEN IMAGE FEATURES IN I_{n-1} AND IMAGE FEATURES IN I_n
65
IDENTIFY 3-D MODEL CO-ORDINATES FOR MATCHED 2-D IMAGE CO-ORDINATES IN I_n TO OBTAIN A SET OF 3D-2D CO-ORDINATES
66
CALCULATE PROVISIONAL RESULT FOR BEST CAMERA SOLUTION FOR Ln USING 3-D TO 2-D SOLVING PROCESS
67
USE PROVISIONAL RESULT FOR BEST CAMERA SOLUTION TO MAP MODEL DATA INTO I_n TO OBTAIN 2-D REFERENCE POINTS
n = n+1
68
PERFORM CONSTRAINED MATCHING IN I_n TO MATCH THE 2-D REFERENCE POINTS WITH ADDITIONAL FEATURES OF I_n
69
IDENTIFY 3-D MODEL CO-ORDINATES AND FORM ENLARGED SET OF MATCHED 2-D IMAGE CO-ORDINATES IN I_n
610
CALCULATE REVISED RESULT FOR CAMERA SOLUTION FOR Ln USING 3-D TO 2-D SOLVING PROCESS WITH ENLARGED SET
611
ANY MORE IMAGES?
YES
NO
A

FIG. 6B

ADDITIONAL STEPS

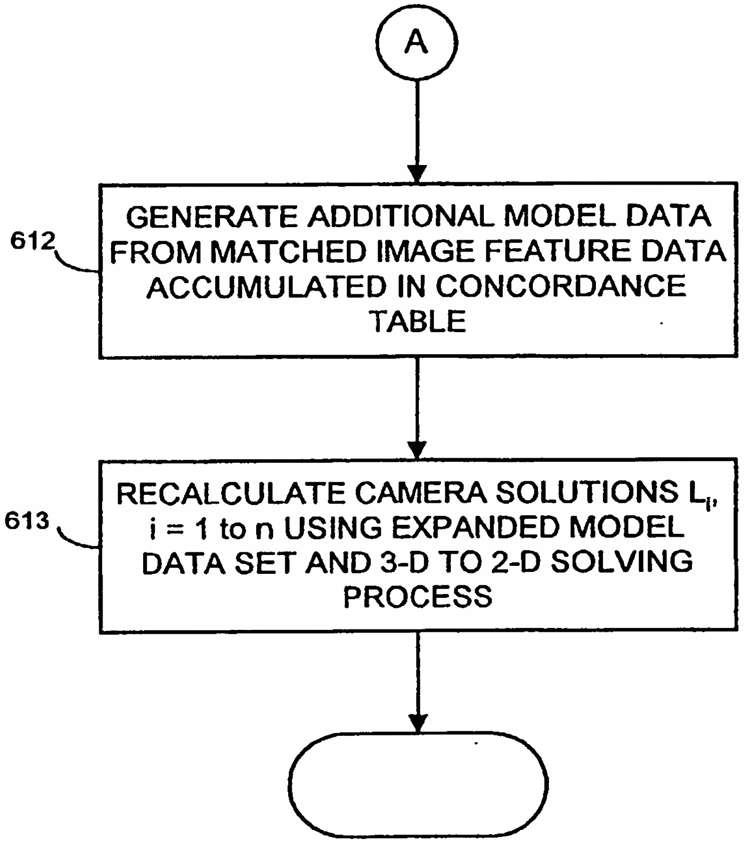
A
612
GENERATE ADDITIONAL MODEL DATA FROM MATCHED IMAGE FEATURE DATA ACCUMULATED IN CONCORDANCE TABLE
613
RECALCULATE CAMERA SOLUTIONS L_i, i = 1 to n USING EXPANDED MODEL DATA SET AND 3-D TO 2-D SOLVING PROCESS

FIG. 7

PERFORMING MATCHING FROM USER INPUT BETWEEN IMAGE FEATURES IN I_{n-1} AND I_n (STEP 64)

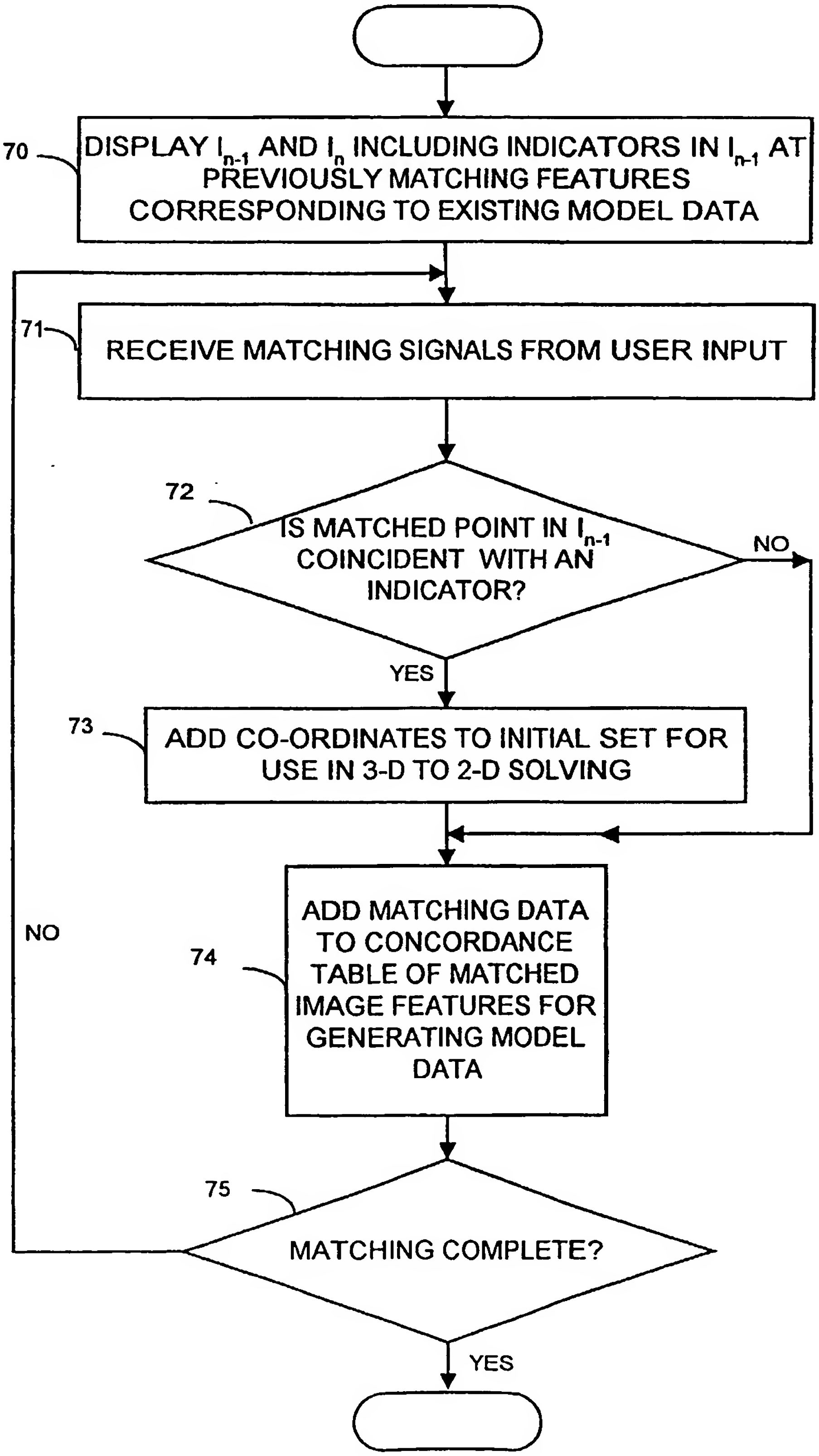
70
DISPLAY I_{n-1} AND I_n INCLUDING INDICATORS IN I_{n-1} AT PREVIOUSLY MATCHING FEATURES CORRESPONDING TO EXISTING MODEL DATA
71
RECEIVE MATCHING SIGNALS FROM USER INPUT
72
IS MATCHED POINT IN I_{n-1} COINCIDENT WITH AN INDICATOR?
NO
YES
73
ADD CO-ORDINATES TO INITIAL SET FOR USE IN 3-D TO 2-D SOLVING
NO
74
ADD MATCHING DATA TO CONCORDANCE TABLE OF MATCHED IMAGE FEATURES FOR GENERATING MODEL DATA
75
MATCHING COMPLETE?
YES

Fig 8

3-D TO 2-D SOLVING PROCESS (STEPS 66 AND 610)

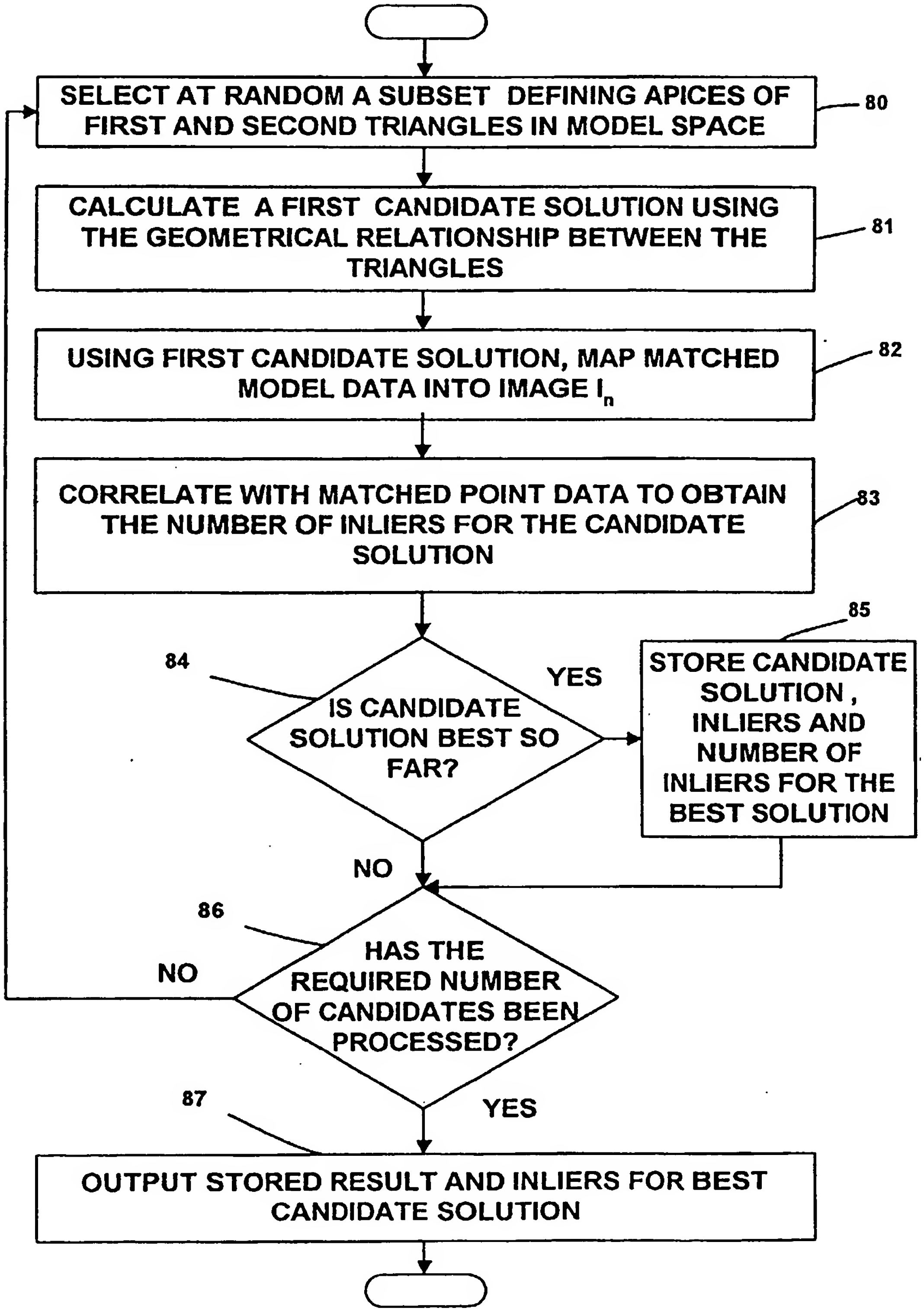
SELECT AT RANDOM A SUBSET DEFINING APICES OF FIRST AND SECOND TRIANGLES IN MODEL SPACE
80
CALCULATE A FIRST CANDIDATE SOLUTION USING THE GEOMETRICAL RELATIONSHIP BETWEEN THE TRIANGLES
81
USING FIRST CANDIDATE SOLUTION, MAP MATCHED MODEL DATA INTO IMAGE I_n
82
CORRELATE WITH MATCHED POINT DATA TO OBTAIN THE NUMBER OF INLIERS FOR THE CANDIDATE SOLUTION
83
85
84
IS CANDIDATE SOLUTION BEST SO FAR?
YES
STORE CANDIDATE SOLUTION , INLIERS AND NUMBER OF INLIERS FOR THE BEST SOLUTION
NO
86
NO
HAS THE REQUIRED NUMBER OF CANDIDATES BEEN PROCESSED?
87
YES
OUTPUT STORED RESULT AND INLIERS FOR BEST CANDIDATE SOLUTION

Fig 9

CALCULATING CANDIDATE 3D - 2D CAMERA SOLUTION

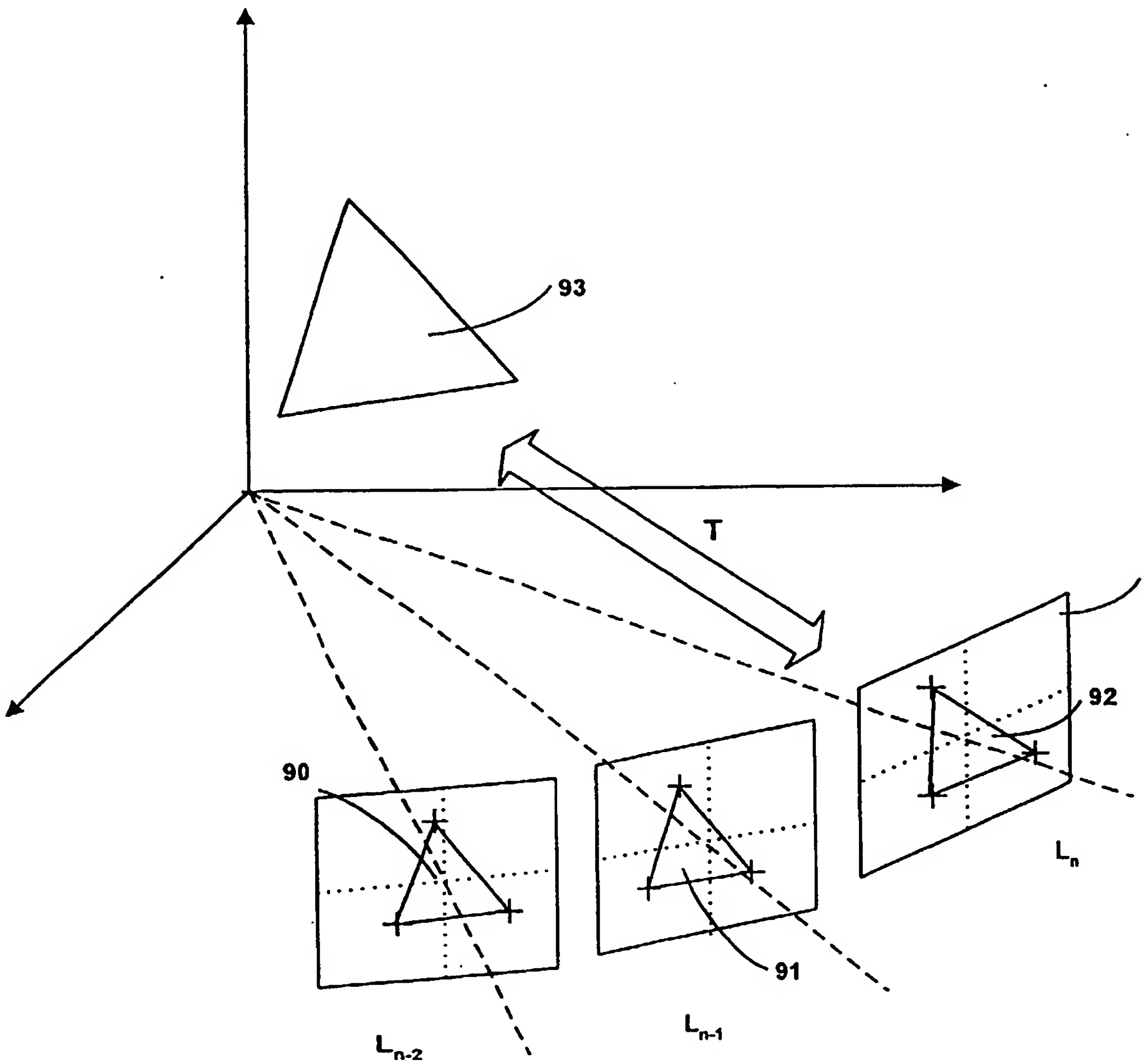

Fig 10

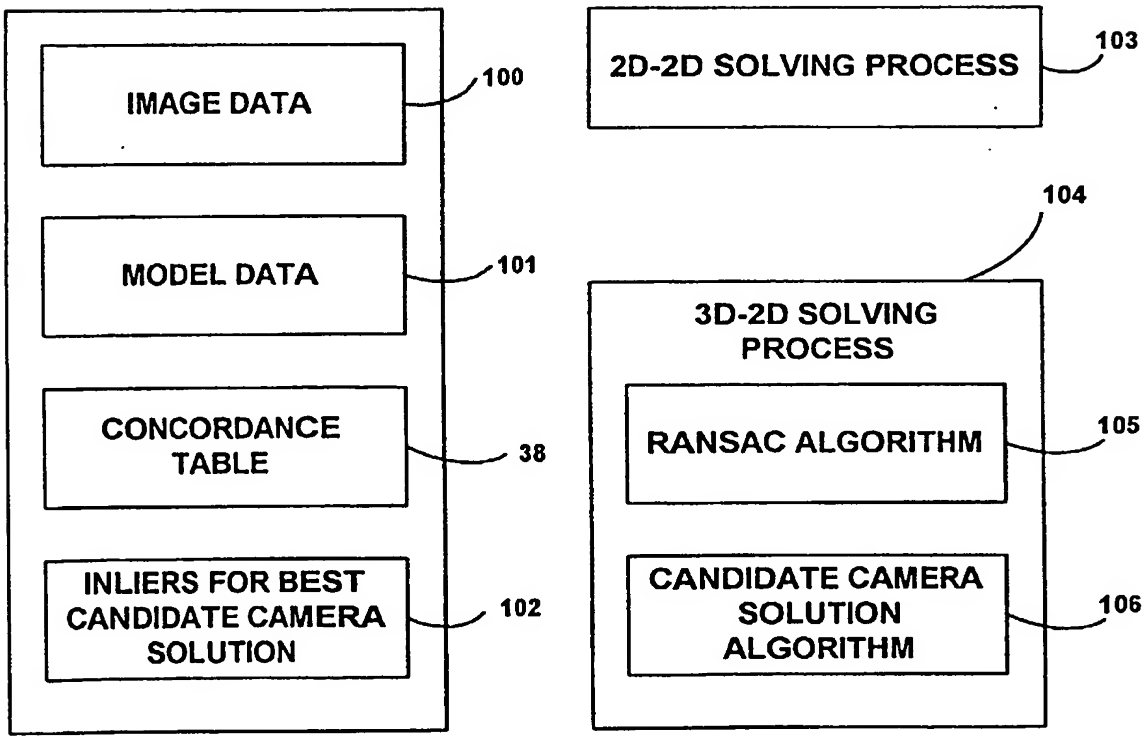
IMAGE DATA
100
MODEL DATA
101
CONCORDANCE TABLE
38
INLIERS FOR BEST CANDIDATE CAMERA SOLUTION
102
2D-2D SOLVING PROCESS
103
104
3D-2D SOLVING PROCESS
RANSAC ALGORITHM
105
CANDIDATE CAMERA SOLUTION ALGORITHM
106

2365243

IMAGE PROCESSING APPARATUS

This invention relates to an apparatus and method of operation of a processor for generating model data for a model in a three-dimensional space from image data representative of a set of camera images of an object.

It is known from EP-A-0898245 to process images of the object taken from different, unknown positions using a matching process in which points in different images which correspond to the same point of the actual object are matched, the matching points being used to determine the relative positions and orientations of cameras from which the images were taken and to then generate model data. This process of determining the camera positions is referred to as calculating a camera solution and EP-A-0898245 discloses a camera solution process relying upon epipolar geometry between virtual image planes of cameras at camera positions from which corresponding images were obtained.

Having solved the camera positions and orientations for an initial three cameras corresponding to an initial three images in a sequence of camera images using a first solution algorithm, EP-A-0898245 teaches that each new image of the sequence of images requires its camera solution to be obtained using a second camera solution algorithm which assumes the camera solution for the preceding image in the sequence to be accurately known from previous calculations. Matching points between the new image and the preceding images in the sequence may then be processed to accumulate further model data.

This known method of camera solution, referred to below as a 2-D to 2-D camera solution process, effectively takes as a starting point pairs of co-ordinates in virtual image planes of a pair of virtual cameras in the three-dimensional model space and calculates the parameters defining the position and orientation of each camera based on these pairs of two-dimensional image co-ordinates for matching points.

It is an object of the present invention to provide an apparatus and method for model generation in which the camera solution process relating to the addition of each new image is improved.

According to the present invention there is disclosed an apparatus and method for generating model data without relying solely upon the 2-D to 2-D camera solution process. Once an initial sequence of images is processed and initial model data generated, camera solutions for subsequent images are calculated by a different process which utilises the model data.

Preferred embodiments of the present invention will now be described by way of example only and with reference to the accompanying drawings of which;

Figure 1 schematically shows the components of a modular system in which the present invention may be embodied;

Figure 2A is a schematic illustration of apparatus in accordance with the present invention;

Figure 2B is a schematic diagram showing the functional components of the apparatus of Figure 2A;

Figure 3A is a schematic diagram showing actual camera positions relative to the object;

Figure 3B is a schematic diagram showing virtual camera positions relative to the model;

Figure 4 is a diagram illustrating a display screen in which camera images are displayed for matching;

Figure 5 is a schematic diagram illustrating the mapping of model points into a virtual image plane of a camera;

Figure 6A and 6B is a schematic flowchart illustrating the overall process for generating model data and calculating camera solutions;

Figure 7 is a flowchart illustrating the matching process enabling a provisional camera solution for a new image to be performed;

Figure 8 is a flowchart illustrating operation of a 3D to 2D solving process;

Figure 9 is a schematic diagram of triangles of selected points used in calculating candidate camera solutions in the process of Figure 8; and

Figure 10 is a schematic diagram of software modules.

Figure 1 schematically shows the components of a modular system in which the present invention may be embodied.

These components can be effected as processor-implemented instructions, hardware or a combination thereof.

Referring to Figure 1, the components are arranged to process data defining images (still or moving) of one or more objects in order to generate data defining a three-dimensional computer model of the object(s).

The input image data may be received in a variety of ways, such as directly from one or more digital cameras, via a storage device such as a disk or CD ROM, by digitisation of photographs using a scanner, or by downloading image data from a database, for example via a datalink such as the Internet, etc.

The generated 3D model data may be used to: display an image of the object(s) from a desired viewing position; control manufacturing equipment to manufacture a model of the object(s), for example by controlling cutting apparatus to cut material to the appropriate dimensions; perform processing to recognise the object(s), for example by comparing it to data stored in a database; carry out processing to measure the object(s), for example by taking absolute measurements to record the size of the object(s), or by comparing the model with models of the object(s) previously generated to determine changes therebetween; carry out processing so as to control a robot to navigate around the object(s); store information in a geographic information system (GIS) or other topographic database; or transmit the object data representing the model to a remote processing device for any such processing, either on a storage device or as a signal (for example, the data may be transmitted in virtual reality modelling language (VRML) format over the Internet, enabling it to be processed by a WWW browser);

etc.

The feature detection and matching module 2 is arranged to receive image data recorded by a still camera from different positions relative to the object(s) (the different positions being achieved by moving the camera and/or the object(s)). The received data is then processed in order to match features within the different images (that is, to identify points in the images which correspond to the same physical point on the object(s)).

The feature detection and tracking module 4 is arranged to receive image data recorded by a video camera as the relative positions of the camera and object(s) are changed (by moving the video camera and/or the object(s)). As in the feature detection and matching module 2, the feature detection and tracking module 4 detects features, such as corners, in the images. However, the feature detection and tracking module 4 then tracks the detected features between frames of image data in order to determine the positions of the features in other images.

The camera position calculation module 6 is arranged to use the features matched across images by the feature detection and matching module 2 or the feature detection and tracking module 4 to calculate the transformation between the camera positions at which the images were recorded and hence determine the orientation and position of the camera focal plane when each image was recorded.

The feature detection and matching module 2 and the camera position calculation module 6 may be arranged to perform processing in an iterative manner. That is, using camera positions and orientations calculated by the camera position calculation module 6, the feature detection and matching module 2 may detect and match further features in the images using epipolar geometry in a conventional manner, and the further matched features may then be used by the camera position calculation module 6 to recalculate the camera positions and orientations.

If the positions at which the images were recorded are already known, then, as indicated by arrow 8 in Figure 1, the image data need not be processed by the feature detection and matching module 2, the feature detection and tracking module 4, or the camera position calculation module 6. For example, the images may be recorded by mounting a number of cameras on a calibrated rig arranged to hold the cameras in known positions relative to the object(s).

Alternatively, it is possible to determine the positions of a plurality of cameras relative to the object(s) by adding calibration markers to the object(s) and calculating the positions of the cameras from the positions of the calibration markers in images recorded by the cameras. The calibration markers may comprise patterns of light projected onto the object(s). Camera calibration module 10 is therefore provided to receive image data from a plurality of cameras at fixed positions showing the object(s) together with calibration markers, and to process the data to determine the positions of the cameras. A preferred method of calculating the positions of the cameras (and also internal parameters of each camera, such as the focal length etc) is described in "Calibrating and 3D Modelling with a Multi-Camera System" by Wiles and Davison in 1999 IEEE Workshop on Multi-View Modelling and Analysis of Visual Scenes, ISBN 0769501109.

The 3D object surface generation module 12 is arranged to receive image data showing the object(s) and data defining the positions at which the images were recorded, and to process the data to generate a 3D computer model representing the actual surface(s) of the object(s), such as a polygon mesh model.

The texture data generation module 14 is arranged to generate texture data for rendering onto the surface model produced by the 3D object surface generation module 12. The texture data is generated from the input image data showing the object(s).

Techniques that can be used to perform the processing in the modules shown in Figure 1 are described in EP-A-0898245, EP-A-0901105, pending US applications 09/129077, 09/129079 and 09/129080, the full contents of which are incorporated herein by cross-reference, and also Annex A.

The present invention may be embodied in particular as part of the camera position calculation module 6.

Figures 2A and 2B illustrate apparatus for use in carrying out the present invention, the apparatus being in the form of a desk top computer having a processor 24 with associated random access memory 35 and mass storage memory 36. Figure 2A illustrates a display monitor 20 which is controlled by the processor 24 and comprises a display screen 21 for the display of images and for use in interactively controlling the processor in generating the model as described below. The random access memory 35 includes a concordance table 38 described below.

A computer mouse 26 used in conjunction with a displayed cursor provides pointing signals 25 in a conventional manner and a keyboard 27 is also provided for the input of user data.

Software for operating the processor 24 may be input to the processor 24 from a portable storage medium in the form of a floppy disc 28 via a disc drive 29.

A modem 22 is also connected to the processor 24 for the input of signals 23 carrying program code or data transmitted over a network such as the internet.

Images I_n (n = 1 to N) in the form of files of image data are input to the processor 24 by connecting a digital camera 30 to an input port 37 of the processor 24.

Figure 3A illustrates the actual positions 30_n of the camera 30 at which successive images in an ordered sequence (n = 1 to N) are taken of an object 31. The sequence is ordered such that, when viewed in plan view from above, the successive positions of the camera 30 move in a progressively anticlockwise direction relative to the object 31.

Figure 3B shows the model 110 in the three-dimensional space of the model and virtual cameras L_n (n = 1 to N), each virtual camera L_n being represented by a respective centre of projection C_n and a virtual image plane 32 spaced from the centre of projection by the focal length of the camera.

The actual positions 30n of the camera 30 in Figure 3A will not in general be known and are therefore calculated by the camera position calculation module 6 from an analysis of the images themselves. An initial camera solution, i.e. calculation of the position and orientation of the virtual cameras L_n relative to the model 110 in the co-ordinate system of the model as shown in Figure 3B, is performed for the initial three camera images I_1, I_2, I_3 to obtain solutions for virtual cameras L_1, L_2 and L_3. To perform the calculation, it is necessary to identify matching points in images I_1 and I_2 and to identify corresponding pairs of matching points in images I_2 and I_3, thereby establishing data in the concordance table 38 of matching points across three images. The camera solution is then calculated using a process hereafter referred to as a 2-D to 2-D process which utilises epipolar geometry, i.e. based on the positions of the matched points in the two-dimensional images when mapped onto the virtual image planes 32 in order to deduce the camera transformation.

A set of model coordinates representative of model points correspond to image points for the matched two dimensional coordinates is then calculated on the basis of the camera solution and entered in the concordance table 38.

Once an initial camera solution from the first triplet of images I_1, I_2, I_3 has been calculated, a different solving process is adopted for subsequent virtual cameras L_n ($n>3$) derived from subsequent images I_n in the sequence. This process utilises the information in the concordance table 38 to identify new matching points found in each new image with coordinates of the existing model data. The camera solution for the new camera is then calculated based on a set of three dimensional model coordinates and corresponding two dimensional image coordinates in the new image. This process is referred to below as a 3-D to 2-D process.

In the solving process, the assumption is made of the camera being representable by a pinhole camera model and that the internal camera parameters of the camera are known.

The overall process of building the model data and performing the camera solutions for a set of images will now be described with reference to the flowchart of Figures 6A and 6B. At step 60, the user selects the 2-D to 2-D camera solution process by selecting the appropriate mode selecting icon 48 as illustrated in Figure 4 and performs matching between the first triplet of images, I_1, I_2 and I_3. This matching process involves the display of pairs of images for inspection by the user who then selects matching pairs by using the mouse 26 and cursor 42 to select matching features in each of the pair of images. When the user has finished matching, the user terminates the matching step by the input of a predetermined control command.

At step 61, the processor 24 calculates the camera solution for the triplet of initial virtual cameras L_1, L_2 and L_3, using the 2-D to 2-D solving process, thereby calculating the position of the respective image plane and look direction for each of the three virtual cameras in the three dimensional space of the model.

At step 62, the processor 24 calculates model data in three dimensions from the measured co-ordinates of matching features established for the initial triplet of images and stores the results with the matching feature data in the concordance table 38. The concordance table then contains an accumulation of data in which the two dimensional coordinates of matching image points are related to the three dimensional co-ordinates of model points.

At step 63, the processor 24 displays a new image I_n (in this case n = 4) for matching with the preceding image I_{n-1} and prompts the user to perform matching at step 64 between the new image I_n and the preceding image I_{n-1}.

This matching process is illustrated in Figure 4 which illustrates the display screen 21 where images I_n and I_{n-1} are displayed for comparison in respective image windows 40 and 41.

Figure 4 also illustrates a row of mode selecting icons 48 which, as mentioned above, may be selected using the cursor 42 and mouse 26 in order to select the various modes of operation adopted by the processor 24 in the modelling and camera solving processes.

At step 64, the user enters co-ordinates of pairs of matching image points and the processor 24 performs matching between the new image I_n and previous image I_{n-1} in a manner which is shown in greater detail in the flowchart of Figure 7. At step 70 of Figure 7, the processor 24 controls the display 20 to display the images I_n and I_{n-1} including indicators 43 in the image I_{n-1} which identify previously matched image points for which existing model data is stored in the concordance table. The user enters co-ordinates of matching image points by using the mouse 26 to move the cursor 42 between the displayed images and select matching features. In some cases, the resulting selection signals 25 received by the processor 24 at step 71 will be determined at step 72 to define a matching pair of points which include a point in I_{n-1} coincident with one of the indicators 43, such matching points being entered at step 73 into an initial set of two dimensional coordinate data to be used in the 3-D to 2-D solving process. The matching data obtained in the matching step 71 is entered at step 74 into the concordance table 38 for use in generating further model data.

The remaining matched points which at step 72 are determined to relate to features in I_{n-1} not previously matched are also added at step 74 as new entries in the concordance table of matched image features to be available for subsequent use in generating further model data.

When at step 75 the matching process is determined to have been terminated by the user inputting a predetermined control command, the processor 24 then begins to process the initial set of two dimensional coordinate data. Referring to Figure 6A, the processor 24 at step 65 begins by identifying the three dimensional model coordinates corresponding to each of the two dimensional image coordinates for the new image I_n in the initial set by referring to the concordance table 38 of matched image features and model data.

The camera solution for the virtual camera L_n is then calculated at step 66 using the 3-D to 2-D solving process, the result being regarded as a provisional result since it is based on the initial set of data which is limited in size by the number of indicators displayed in the previous image I_{n-1}. In order to make full use of all of the existing three dimensional model data, the processor 24 at step 67 maps the three dimensional model points represented by the remainder of the set of model data into the two dimensional virtual image plane of the virtual camera L_n, thereby obtaining a set of two dimensional reference points in the image plane 52.

Figure 5 illustrates this mapping process schematically where a small set of three dimensional model coordinates 50 are illustrated as being mapped into a corresponding set of two-dimensional reference points 51 in the image plane 52 of camera L_n.

At step 68, the processor 24 performs automatic matching of features in the new image I_n with the reference points 51 obtained in step 67 using a constrained matching technique in which the search for a matching feature to each of the reference points is confined to a localised area proximate to the reference point in the new image.

After completing the constrained matching process, the processor 24 at step 69 is then able to identify an enlarged set of two dimensional image coordinates in the new image I_n for which correspondence is matched with three dimensional model coordinates, including the results of both step 68 and step 65.

A revised result for the camera solution for the virtual camera L_n is then calculated by again using the 3-D to 2-D solving process but based on the enlarged set of 2-D matched coordinates and corresponding 3-D model data at step 610.

If at step 611 the processor 24 determines that there are more images to be processed, the process repeats from step 63 for a new image I_n for which the value of n is incremented by 1.

When all of the images have been processed, additional model data is calculated at step 612 of Figure 6B using all of the matched image feature data accumulated during each performance of the matching process of step 64 and the automatic matching process of step 68 for all of the images, provided that matching of a feature in at least three images is required before a new model data point can be determined.

Using the expanded model data set established in step 612, the processor 24 at step 613 applies the 3-D to 2-D solving process to each of the virtual cameras L_n in order to refine the camera solutions for use in any subsequent processing.

The 3D to 2D solving process used in steps 66 and 610 will now be described with reference to Figure 8. For this example, the use of the 3D to 2D process of step 66 is described for camera L_n where n is greater than 3. As shown in Figure 9, the solution for camera L_n requires a set of coordinates for matching points in each of cameras L_n, L_{n-1} and L_{n-2} where cameras L_{n-1} and L_{n-2} already have known position and orientation as a result of earlier solving processes.

Each pair of matching points in L_{n-1} and L_{n-2} has a corresponding three-dimensional model point in the existing model data, the association between these sets of data being defined in the concordance table 38.

For each pair of matching image points represented in the image data for L_{n-1} and L_{n-2} there is a matching image point represented in the image data for camera L_n as a result of the matching process performed in step 64 referred to above.

Reference will be made to the method steps of Figure 8 as well as the diagram of Figure 9 in the following description. The processor in implementing the steps of

Figure 8 uses a RANSAC (random sampling and consensus) algorithm. At step 80, the processor 24 selects at random three matches between images I_n, I_{n-1} and I_{n-2}, such that each match comprises sets of two-dimensional image coordinates expressed in pixel numbers. These three matches have coordinates which define the apices of respective imaginary triangles 90, 91 and 92 as shown in Figure 9. The corresponding three-dimensional coordinates in the model data define model points at apices of a further imaginary triangle 93 whose positions are known in "world coordinates" or in other words relative to the frame of reference with which the model data is defined. The triangle 92 of image points in the new image I_n may therefore be regarded as a two-dimensional projection of the triangle 93 of model points onto the virtual image plane 52 of the camera L_n so that the position and orientation of the image plane 52 can be calculated using a standard geometrical transformation represented in Figure 9 by arrow T.

The result of this calculation will be a set of values defining the position in world coordinates and the orientation relative to the model frame of reference of the image plane 52 and constitutes a first candidate solution for the required camera solution for L_n.

As shown in Figure 8, step 81 of calculating this first candidate solution is followed by step 82 of using the first candidate solution to map all of the model points corresponding to the initial set of image points into the image plane I_n. If the first candidate solution were in fact a perfect solution, the mapped points would be expected to substantially coincide with the user entered matched image points. In practice, however, the mapped points will be displaced relative to the matched image points by a number of pixels which provides a measure of the degree of correlation between the mapped points and matched image points.

At step 83, a correlation calculation is performed between the mapped points and the matched image points by counting the number of mapped points which fall within a predetermined number of pixels radius of the matched image points. In this example, the predetermined number of pixels is three.

The number of matching pairs of mapped points and matched image points in the image is equal to the number of inliers for this candidate solution, each inlier comprising data defining co-ordinates of a model point together with co-ordinates of corresponding image points in each of at least three images.

The above calculation is repeated for a number of further candidate solutions and at step 84 the processor 24 determines whether the current candidate solution produces the best result so far in terms of a number of inliers. If so, the candidate solution and number of inliers is stored in step 85 as the result of the process.

At step 86, it is determined whether the required number of candidate solutions has yet been processed, and if not, the process repeats from step 80 where a new set of three matches are selected at random and the above described steps repeated.

When the required number of candidate solutions has been processed, the processor outputs at step 87 the stored result in terms of the candidate solution and number of inliers stored in step 85 for the optimum candidate solution. Also output are the inliers for the candidate solution in terms of the set of point matches verified by the solving process to represent consistent matched data across the three images I_n, I_{n-1} and I_{n-2}.

The calculation referred to above at step 81 makes use of the well-known projection geometry described for example in "Computer and Robot Vision, Volume 2" by Robert M Haralick and Linda G Shapiro, 1993, Addison Wesley, pages 85 to 91. This publication describes in this passage a transformation which may readily be inverted to suit the calculation required for the present context and defining thereby the transformation T referred to above.

Figure 10 shows schematically some of the software modules utilised in the above process. An image data file 100 contains image data input from a camera or the like and a model data file 101 contains the model data generated from the image data.

Concordance table 38 referred to above includes related entries identifying the correspondence between matched image data in two or more images and the corresponding model data co-ordinates.

An inliers file 102 contains information defining the inliers found in each of the best candidate camera solutions and represents a set of point matches which are correct and verified to be consistent across three or more images.

The data files 100, 101, 38 and 102 are typically held in random access memory 35 during processing and ultimately stored in mass storage memory 36 of Figure 2.

Also shown in Figure 10 are the processing elements including the 2-D to 2-D solving process 103 and the 3-D to 2-D solving process 104 which includes both the RANSAC algorithm 105 and the candidate camera solution 1 algorithm 106.

The RANSAC algorithm 105 and candidate camera solution algorithm 106 constitute computer programs comprising processor implementable instructions which may be stored in a storage medium such as floppy disc 28 or may be downloaded as signals 23 from a network such as the internet. Such signals and storage mediums embodying these instructions therefore constitute aspects of the present invention. Similarly, other programs for carrying out the above described embodiments including control software for controlling operation of the above software modules may be stored in the storage medium or transmitted as a signal, thereby constituting further aspects of the present invention.

ANNEX A

1. CORNER DETECTION

1.1 Summary

This process described below calculates corner points, to sub-pixel accuracy, from a single grey scale or colour image. It does this by first detecting edge boundaries in the image and then choosing corner points to be points where a strong edge changes direction rapidly. The method is based on the facet model of corner detection, described in Haralick and Shapiro[1].

1.2 Algorithm

The algorithm has four stages:

(1) Create grey scale image (if necessary);
(2) Calculate edge strengths and directions;
(3) Calculate edge boundaries;
(4) Calculate corner points.

1.2.1 Create grey scale image

The corner detection method works on grey scale images. For colour images, the colour values are first converted to floating point grey scale values using the formula:

$$grey\_scale = (0.3 \times red) + (0.59 \times green) + (0.11 \times blue) \quad \ldots.A\text{-}1$$

This is the standard definition of brightness as defined by NTSC and described in Foley and van Dam[ii].

1.2.2 Calculate edge strengths and directions

The edge strengths and directions are calculated using the 7x7 integrated directional derivative gradient operator discussed in section 8.9 of Haralick and Shapiro[i].

The row and column forms of the derivative operator are both applied to each pixel in the grey scale image. The results are combined in the standard way to calculate the edge strength and edge direction at each pixel.

The output of this part of the algorithm is a complete derivative image.

1.2.3 Calculate edge boundaries

The edge boundaries are calculated by using a zero crossing edge detection method based on a set of 5x5 kernels describing a bivariate cubic fit to the neighbourhood of each pixel.

The edge boundary detection method places an edge at all pixels which are close to a negatively sloped zero crossing of the second directional derivative taken in the direction of the gradient, where the derivatives are defined using the bivariate cubic fit to the grey level surface. The subpixel location of the zero crossing is also stored along with the pixel location.

The method of edge boundary detection is described in more detail in section 8.8.4 of Haralick and Shapiro[1].

1.2.4 Calculate corner points

The corner points are calculated using a method which uses the edge boundaries calculated in the previous step.

Corners are associated with two conditions:

(1) the occurrence of an edge boundary; and

(2) significant changes in edge direction.

Each of the pixels on the edge boundary is tested for "cornerness" by considering two points equidistant to it along the tangent direction. If the change in the edge direction is greater than a given threshold then the point is labelled as a corner. This step is described in section 8.10.1 of Haralick and Shapiro[1].

Finally the corners are sorted on the product of the edge strength magnitude and the change of edge direction. The top 200 corners which are separated by at least 5 pixels are output.

2. FEATURE TRACKING

2.1 Summary

This process described below tracks feature points (typically corners) across a sequence of grey scale or colour images.

The tracking method uses a constant image velocity Kalman filter to predict the motion of the corners, and a correlation based matcher to make the measurements of corner correspondences.

The method assumes that the motion of corners is smooth enough across the sequence of input images that a constant velocity Kalman filter is useful, and that corner measurements and motion can be modelled by gaussians.

2.2 Algorithm

1) Input corners from an image.

2) Predict forward using Kalman filter.

3) If the position uncertainty of the predicted corner is greater than a threshold, Δ, as measured by the state positional variance, drop the corner from the list of currently tracked corners.

4) Input a new image from the sequence.

5) For each of the currently tracked corners:
   a) search a window in the new image for pixels which match the corner;
   b) update the corresponding Kalman filter, using any new observations (i.e. matches).

6) Input the corners from the new image as new points to be tracked (first, filtering them to remove any which are too close to existing tracked points).

7) Go back to (2)

2.2.1 Prediction

This uses the following standard Kalman filter equations for prediction, assuming a constant velocity and random uniform gaussian acceleration model for the dynamics:

$$X_{n+1} = \Theta_{n+1,n} X_n \qquad \text{....A-2}$$

$$K_{n+1} = \Theta_{n+1,n} K_n \Theta^T_{n+1,n} + Q_n \qquad \text{....A-3}$$

where x is the 4D state of the system, (defined by the position and velocity vector of the corner), K is the state covariance matrix, Θ is the transition matrix, and Q is the process covariance matrix.

In this model, the transition matrix and process covariance matrix are constant and have the following values:

$$\Theta_{n+1,n} = \begin{pmatrix} I & I \\ 0 & I \end{pmatrix} \qquad \text{....A-4}$$

$$Q_n = \begin{pmatrix} 0 & 0 \\ 0 & \sigma_v^2 I \end{pmatrix} \qquad \text{....A-5}$$

2.2.2 Searching and matching

This uses the positional uncertainty (given by the top two diagonal elements of the state covariance matrix, K) to define a region in which to search for new measurements (i.e. a range gate).

The range gate is a rectangular region of dimensions:

$$\Delta x = \sqrt{K_{11}},\ \Delta y = \sqrt{K_{22}} \qquad \text{....A-6}$$

The correlation score between a window around the previously measured corner and each of the pixels in the range gate is calculated.

The two top correlation scores are kept.

If the top correlation score is larger than a threshold, C_0, and the difference between the two top correlation scores is larger than a threshold ΔC, then the pixel with the top correlation score is kept as the latest measurement.

2.2.3 Update

The measurement is used to update the Kalman filter in the standard way:

$$G = KH^T(HKH^T+R)^{-1} \qquad \text{....A-7}$$

$$x \rightarrow x+G(\hat{x}-Hx) \qquad \text{....A-8}$$

$$K \rightarrow (I-GH)K \qquad \text{....A-9}$$

where G is the Kalman gain, H is the measurement matrix, and R is the measurement covariance matrix.

In this implementation, the measurement matrix and measurement covariance matrix are both constant, being given by:

$$H = (I\ 0) \qquad \text{....A-10}$$

$$R = \sigma^2 I \qquad \text{....A-11}$$

2.2.4 Parameters

The parameters of the algorithm are:

Initial conditions: x_0 and K_0.

Process velocity variance: σ_v^2.

Measurement variance: σ^2.

Position uncertainty threshold for loss of track: Δ.

Covariance threshold: C_0.

Matching ambiguity threshold: ΔC.

For the initial conditions, the position of the first corner measurement and zero velocity are used, with an initial covariance matrix of the form:

$$K_0 = \begin{pmatrix} 0 & 0 \\ 0 & \sigma_0^2 I \end{pmatrix} \qquad \text{....A-12}$$

σ_0^2 is set to σ_0^2 = 200(pixels/frame)^2.

The algorithm's behaviour over a long sequence is anyway not too dependent on the initial conditions.

The process velocity variance is set to the fixed value of 50 (pixels/frame)^2. The process velocity variance would have to be increased above this for a hand-held sequence. In fact it is straightforward to obtain a reasonable value for the process velocity variance adaptively.

The measurement variance is obtained from the following model:

$$\sigma^2 = (rK+a) \qquad \text{....A-13}$$

where K = √(K$_{11}$K$_{22}$) is a measure of the positional uncertainty, r is a parameter related to the likelihood of obtaining an outlier, and a is a parameter related to the measurement uncertainty of inliers. "r" and "a" are set to r=0.1 and a=1.0.

This model takes into account, in a heuristic way, the fact that it is more likely that an outlier will be obtained if the range gate is large.

The measurement variance (in fact the full measurement covariance matrix R) could also be obtained from the behaviour of the auto-correlation in the neighbourhood of the measurement. However this would not take into account the likelihood of obtaining an outlier.

The remaining parameters are set to the values: $\Delta=400$ pixels^2, $C_0=0.9$ and $\Delta C=0.001$.

3. 3D SURFACE GENERATION

3.1 Architecture

In the method described below, it is assumed that the object can be segmented from the background in a set of images completely surrounding the object. Although this restricts the generality of the method, this constraint can often be arranged in practice, particularly for small objects.

The method consists of five processes, which are run consecutively:

- First, for all the images in which the camera positions and orientations have been calculated, the object is segmented from the background, using colour information. This produces a set of binary images, where the pixels are marked as being either object or background.

- The segmentations are used, together with the camera positions and orientations, to generate a voxel carving, consisting of a 3D grid of voxels enclosing the object. Each of the voxels is marked as being either object or empty space.

- The voxel carving is turned into a 3D surface triangulation, using a standard triangulation algorithm (marching cubes).

- The number of triangles is reduced substantially by passing the triangulation through a decimation process.

- Finally the triangulation is textured, using appropriate parts of the original images to provide the texturing on the triangles.

3.2 Segmentation

The aim of this process is to segment an object (in front of a reasonably homogeneous coloured background) in an image using colour information. The resulting binary image is used in voxel carving.

Two alternative methods are used:

Method 1: input a single RGB colour value representing the background colour - each RGB pixel in the image is examined and if the Euclidean distance to the background colour (in RGB space) is less than a specified threshold the pixel is labelled as background (BLACK).

Method 2: input a "blue" image containing a representative region of the background.

The algorithm has two stages:

(1) Build a hash table of quantised background colours

(2) Use the table to segment each image.

Step 1) Build hash table

Go through each RGB pixel, p, in the "blue" background image.

Set q to be a quantised version of p. Explicitly:

$$q = (p+t/2)/t \qquad \text{....A-14}$$

where t is a threshold determining how near RGB values need to be to background colours to be labelled as background.

The quantisation step has two effects:

1) reducing the number of RGB pixel values, thus increasing the efficiency of hashing;

2) defining the threshold for how close a RGB pixel has to be to a background colour pixel to be labelled as background.

q is now added to a hash table (if not already in the table) using the (integer) hashing function

$$h(q) = (q\_red \;\&\; 7)*2^6+(q\_green \;\&\; 7)*2^3+(q\_blue \;\&\; 7) \qquad \text{....A-15}$$

That is, the 3 least significant bits of each colour field are used. This function is chosen to try and spread out the data into the available bins. Ideally each bin in the hash table has a small number of colour entries. Each quantised colour RGB triple is only added once to the table (the frequency of a value is irrelevant).

Step 2) Segment each image

Go through each RGB pixel, v, in each image.

Set w to be the quantised version of v as before.

To decide whether w is in the hash table, explicitly look at all the entries in the bin with index h(w) and see if any of them are the same as w. If yes, then v is a background pixel - set the corresponding pixel in the output image to BLACK. If no then v is a foreground pixel - set the corresponding pixel in the output image to WHITE

Post Processing: For both methods a post process is performed to fill small holes and remove small isolated regions.

A median filter is used with a circular window. (A circular window is chosen to avoid biasing the result in the x or y directions).

Build a circular mask of radius r. Explicitly store the start and end values for each scan line on the circle.

Go through each pixel in the binary image.

Place the centre of the mask on the current pixel. Count the number of BLACK pixels and the number of WHITE pixels in the circular region.

If (#WHITE pixels ≥ #BLACK pixels) then set corresponding output pixel to WHITE. Otherwise output pixel is BLACK.

3.3 Voxel carving

The aim of this process is to produce a 3D voxel grid, enclosing the object, with each of the voxels marked as either object or empty space.

The input to the algorithm is:

- a set of binary segmentation images, each of which is associated with a camera position and orientation;

- 2 sets of 3D co-ordinates, (xmin, ymin, zmin) and (xmax, ymax, zmax), describing the opposite vertices of a cube surrounding the object;

- a parameter, n, giving the number of voxels required in the voxel grid.

A pre-processing step calculates a suitable size for the voxels (they are cubes) and the 3D locations of the voxels, using n, (xmin, ymin, zmin) and (xmax, ymax, zmax).

Then, for each of the voxels in the grid, the mid-point of the voxel cube is projected into each of the segmentation images. If the projected point falls onto a pixel which is marked as background, on any of the images, then the corresponding voxel is marked as empty space, otherwise it is marked as belonging to the object.

Voxel carving is described further in "Rapid Octree Construction from Image Sequences" by R. Szeliski in CVGIP: Image Understanding, Volume 58, Number 1, July 1993, pages 23-32.

3.4 Marching cubes

The aim of the process is to produce a surface triangulation from a set of samples of an implicit function representing the surface (for instance a signed distance function). In the case where the implicit function has been obtained from a voxel carve, the implicit function takes the value -1 for samples which are inside the object and +1 for samples which are outside the object.

Marching cubes is an algorithm that takes a set of samples of an implicit surface (e.g. a signed distance function) sampled at regular intervals on a voxel grid, and extracts a triangulated surface mesh. Lorensen and Cline[iii] and Bloomenthal[iv] give details on the algorithm and its implementation.

The marching-cubes algorithm constructs a surface mesh by "marching" around the cubes while following the zero crossings of the implicit surface f(x)=0, adding to the triangulation as it goes. The signed distance allows the marching-cubes algorithm to interpolate the location of the surface with higher accuracy than the resolution of the volume grid. The marching cubes algorithm can be used as a continuation method (i.e. it finds an initial surface point and extends the surface from this point).

3.5 Decimation

The aim of the process is to reduce the number of triangles in the model, making the model more compact and therefore easier to load and render in real time.

The process reads in a triangular mesh and then randomly removes each vertex to see if the vertex contributes to the shape of the surface or not. (i.e. if the hole is filled, is the vertex a "long" way from the filled hole). Vertices which do not contribute to the shape are kept out of the triangulation. This results in fewer vertices (and hence triangles) in the final model.

The algorithm is described below in pseudo-code.

```
INPUT
Read in vertices
Read in triples of vertex IDs making up triangles

PROCESSING
Repeat NVERTEX times
    Choose a random vertex, V, which hasn't been
    chosen before
```

```
        Locate set of all triangles having V as a vertex, S
        Order S so adjacent triangles are next to each other
        Re-triangulate triangle set, ignoring V (i.e. remove selected triangles & V and then fill in hole)
        Find the maximum distance between V and the plane of each triangle
        If (distance < threshold)
            Discard V and keep new triangulation
        Else
            Keep V and return to old triangulation

    OUTPUT
    Output list of kept vertices
    Output updated list of triangles
```

The process therefore combines adjacent triangles in the model produced by the marching cubes algorithm, if this can be done without introducing large errors into the model.

The selection of the vertices is carried out in a random order in order to avoid the effect of gradually eroding a large part of the surface by consecutively removing neighbouring vertices.

3.6 Further Surface Generation Techniques

Further techniques which may be employed to generate a 3D computer model of an object surface include voxel colouring, for example as described in "Photorealistic Scene Reconstruction by Voxel Coloring" by Seitz and Dyer in Proc. Conf. Computer Vision and Pattern Recognition 1997, p1067-1073, "Plenoptic Image Editing" by Seitz and Kutulakos in Proc. 6th International Conference on Computer Vision, pp 17-24, "What Do N Photographs Tell Us About 3D Shape?" by Kutulakos and Seitz in University of Rochester Computer Sciences Technical Report 680, January 1998, and "A Theory of Shape by Space Carving" by Kutulakos and Seitz in University of Rochester Computer Sciences Technical Report 692, May 1998.

4. TEXTURING

The aim of the process is to texture each surface polygon (typically a triangle) with the most appropriate image texture. The output of the process is a VRML model of the surface, complete with texture co-ordinates.

The triangle having the largest projected area is a good triangle to use for texturing, as it is the triangle for which the texture will appear at highest resolution.

A good approximation to the triangle with the largest projected area, under the assumption that there is no substantial difference in scale between the different images, can be obtained in the following way.

For each surface triangle, the image "i" is found such that the triangle is the most front facing (i.e. having the greatest value for $\hat{n}_t.\hat{v}_i$, where $\hat{n}_t$ is the triangle normal and $\hat{v}_i$ is the viewing direction for the "i" th camera). The vertices of the projected triangle are then used as texture co-ordinates in the resulting VRML model.

This technique can fail where there is a substantial amount of self-occlusion, or several objects occluding each other. This is because the technique does not take into account the fact that the object may occlude the selected triangle. However, in practice this does not appear to be much of a problem.

It has been found that, if every image is used for texturing then this can result in very large VRML models being produced. These can be cumbersome to load and render in real time. Therefore, in practice, a subset of images is used to texture the model. This subset may be specified in a configuration file.

References

i R M Haralick and L G Shapiro: "Computer and Robot Vision Volume 1", Addison-Wesley, 1992, ISBN 0-201-10877-1 (v.1), section 8.

ii J Foley, A van Dam, S Feiner and J Hughes: "Computer Graphics: Principles and Practice", Addison-Wesley, ISBN 0-201-12110-7.

iii W.E. Lorensen and H.E.Cline: "Marching Cubes: A

High Resolution 3D Surface Construction Algorithm", in Computer Graphics, SIGGRAPH 87 proceedings, 21: 163-169, July 1987.

iv J. Bloomenthal: "An Implicit Surface Polygonizer", Graphics Gems IV, AP Professional, 1994, ISBN 0123361559, pp 324-350.

CLAIMS:

1. A method of creating a 3-D model of an object by processing images taken from a series of respective camera positions relative to the object;

the method comprising;

processing an initial sequence of the images to define respective image co-ordinates of matching features to generate therefrom a set of model data defining model points in a 3-D space of the model and to obtain respective camera solutions representative of positions and orientations of virtual cameras in the 3-D space defining views of the model corresponding to the images; and

adding a new image to the sequence and processing the new image to obtain a camera solution for a corresponding new virtual camera for use in generating further model data;

wherein the processing of the new image comprises;

(a) identifying a plurality of image points in the new image which are matched to a respective plurality of image points of at least one preceding image of the sequence for which respective 3-D model data defining corresponding model points exists;

(b) determining a set of 2-D image co-ordinates of the identified image points in the new image and co-ordinates of respective model points; and

(c) processing the set of 2-D image point co-ordinates and respective 3-D model point co-ordinates to obtain the camera solution for the new image using a solving process in which the position and orientation of an image plane representative of the new virtual camera are calculated from a geometrical relationship in the 3-D model space between model points and image points defined by the set of co-ordinates.

2. A method as claimed in claim 1 wherein the solving process of step (c) comprises:

selecting a subset of image points and corresponding model points defined by the set of co-ordinates;

calculating a candidate camera solution from the geometrical relationship between the points defined by the subset;

repeating the selecting and calculating step for different subsets to obtain successive candidate camera solutions;

evaluating the candidate camera solutions; and

selecting a best candidate camera solution on the basis of the evaluating step.

3. A method as claimed in claim 2 wherein each subset

comprises a selection of three model points and respective image points, the three model points defining apices of a first triangle and the three image points defining apices of a second triangle, and whereby the geometrical relationship is defined by the second triangle constituting a mapping of the first triangle onto the image plane.

4. A method as claimed in claim 3 wherein the mapping comprises a perspective mapping.

5. A method as claimed in any of claims 2 to 4 wherein the evaluating step comprises mapping model points defined by the set of existing model data into the image plane using the candidate solution to obtain coordinates of reference points in the image plane; and

correlating the reference points with the image points.

6. A method as claimed in claim 5 wherein the correlating step comprises determining whether each image point lies within a predetermined number of pixel units from a respective reference point and counting the number of such image points as a measure of correlation.

7. A method as claimed in claim 6, wherein the step of selecting the best candidate camera solution comprises selection according to the highest measure of correlation.

8. A method as claimed in any of claims 2 to 7 comprising determining a set of inliers for the best candidate solution wherein each inlier comprises data defining co-ordinates of a model point together with co-ordinates of corresponding image points in each of at least three images of the sequence.

9. A method as claimed in any of claims 2 to 8, comprising the step of using the best camera solution to project the remainder of the set of existing model data into the image plane of the new virtual camera to obtain a set of further reference points in the image plane;

performing matching using the further reference points to identify matching further image points of the new image; and

adding the co-ordinates of the further image points and respective model points to the set of co-ordinates determined at step (b) to thereby obtain an enlarged set of co-ordinates.

10. A method as claimed in claim 9 comprising processing the enlarged set of co-ordinates using the solving process of step (c) to obtain a revised result for the best camera solution.

11. A method as claimed in claim 10 further comprising generating further model data in accordance with the revised camera solution using image co-ordinates of matching features in the new image and preceding images of the sequence, and adding the further model data to the set of model data to form an expanded set of model data.

12. A method as claimed in claim 11 further comprising repeating the calculation of camera solutions for the sequence of images using the expanded set of model data.

13. Apparatus for creating a 3-D model of an object by processing images taken from a series of respective camera positions relative to the object;

the apparatus comprising;

means for processing an initial sequence of the images to define respective image co-ordinates of matching features to generate therefrom a set of model data defining model points in a 3-D space of the model and to obtain respective camera solutions representative of positions and orientations of virtual cameras in the 3-D space defining views of the model corresponding to the images; and

means for adding a new image to the sequence and processing the new image to obtain a camera solution for a corresponding new virtual camera for use in generating further model data;

wherein the means for processing of the new image comprises;

(a) means for identifying a plurality of image points in the new image which are matched to a respective plurality of image points of at least one preceding image of the sequence for which respective 3-D model data defining corresponding model points exists;

(b) means for determining a set of 2-D image co-ordinates of the identified image points in the new image and co-ordinates of respective model points; and

(c) solving means for processing the set of 2-D image point co-ordinates and respective 3-D model point co-ordinates to obtain the camera solution for the new image using a solving process in which the position and orientation of an image plane representative of the new virtual camera are calculated from a geometrical relationship in the 3-D model space between model points and image points defined by the set of co-ordinates.

14. Apparatus as claimed in claim 13 wherein the solving means comprises:

means for selecting a subset of image points and corresponding model points defined by the set of co-ordinates;

means for calculating a candidate camera solution from the geometrical relationship between the points defined by the subset;

means for repeating the selecting and calculating step for different subsets to obtain successive candidate camera solutions;

means for evaluating the candidate camera solutions; and

means for selecting a best candidate camera solution.

15. Apparatus as claimed in claim 14 wherein each subset comprises a selection of three model points and respective image points, the three model points defining apices of a first triangle and the three image points defining apices of a second triangle, and whereby the geometrical relationship is defined by the second triangle constituting a mapping of the first triangle onto the image plane.

16. Apparatus as claimed in claim 15 wherein the mapping comprises a perspective mapping.

17. Apparatus as claimed in any of claims 14 to 16 wherein the evaluating means comprises means for mapping model points defined by the set of existing model data into the image plane using the candidate solution to obtain coordinates of reference points in the image plane; and

means for correlating the reference points with the image points.

18. Apparatus as claimed in claim 17, wherein the correlating means comprises means for determining whether each image point lies within a predetermined number of pixel units from a respective reference point and counting the number of such image points as a measure of correlation.

19. Apparatus as claimed in claim 18, wherein the means for selecting the best candidate camera solution comprises means for selection according to the highest measure of correlation.

20. Apparatus as claimed in any of claims 14 to 19

comprising means for determining a set of inliers for the best candidate solution wherein each inlier comprises data defining co-ordinates of a model point together with co-ordinates of corresponding image points in each of at least three images of the sequence.

21. Apparatus as claimed in any of claims 14 to 20 wherein the solving means is operable to use the best camera solution to project the remainder of the set of existing model data into the image plane of the new virtual camera to obtain a set of further reference points in the image plane;

to perform matching using the further reference points to identify matching further image points of the new image; and

to add the co-ordinates of the further image points and respective model points to the set of co-ordinates determined at step (b) to thereby obtain an enlarged set of co-ordinates.

22. Apparatus as claimed in claim 20 wherein the solving means is operable to process the enlarged set of co-ordinates to obtain a revised result for the best camera solution.

23. Apparatus as claimed in claim 22 further comprising means for generating further model data in accordance with the revised camera solution using image co-ordinates of matching features in the new image and preceding images of the sequence, and adding the further model data to the set of model data to form an expanded set of model data.

24. Apparatus as claimed in claim 23 further comprising means for repeating the calculation of camera solutions for the sequence of images using the expanded set of model data.

25. In a method of creating a 3-D model of an object by processing images taken from a series of respective camera positions relative to the object;

the method comprising;

processing an initial sequence of the images to define respective image co-ordinates of matching features to generate therefrom a set of model data defining model points in a 3-D space of the model and to obtain respective camera solutions representative of positions and orientations of virtual cameras in the 3-D space defining views of the model corresponding to the images;

an improvement comprising:

adding a new image to the sequence and processing the new image to obtain a camera solution for a corresponding new virtual camera for use in generating further model data;

wherein the processing of the new image comprises;

(a) identifying a plurality of image points in the new image which are matched to a respective plurality of image points of at least one preceding image of the sequence for which respective 3-D model data defining corresponding model points exists;

(b) determining a set of 2-D image co-ordinates of the identified image points in the new image and co-ordinates of respective model points; and

(c) processing the set of 2-D image point co-ordinates and respective 3-D model point co-ordinates to obtain the camera solution for the new image using a solving process in which the position and orientation of an image plane representative of the new virtual camera are calculated from a geometrical relationship in the 3-D model space between model points and image points defined by the set of co-ordinates.

26. In a apparatus for creating a 3-D model of an object by processing images taken from a series of respective camera positions relative to the object;

the apparatus comprising;

means for processing an initial sequence of the images to define respective image co-ordinates of matching features to generate therefrom a set of model data defining model points in a 3-D space of the model and to obtain respective camera solutions representative of positions and orientations of virtual cameras in the 3-D space defining views of the model corresponding to the images;

an improvement comprising:

means for adding a new image to the sequence and processing the new image to obtain a camera solution for a corresponding new virtual camera for use in generating further model data;

wherein the means for processing of the new image comprises;

(a) means for identifying a plurality of image points in the new image which are matched to a respective plurality of image points of at least one preceding image of the sequence for which respective 3-D model data defining corresponding model points exists;

(b) means for determining a set of 2-D image co-ordinates of the identified image points in the new image and co-ordinates of respective model points; and

(c) means for processing the set of 2-D image point co-ordinates and respective 3-D model point co-ordinates to obtain the camera solution for the new image using a solving process in which the position and orientation of an image plane representative of the new virtual camera are calculated from a geometrical relationship in the 3-D model space between model points and image points defined by the set of co-ordinates.

27. In an apparatus for creating a 3-D model of an object by processing images taken from a series of respective camera positions relative to the object;

processing an initial sequence of the images to define respective image co-ordinates of matching features to generate therefrom a set of model data defining model points in a 3-D space of the model and to obtain respective camera solutions representative of positions and orientations of virtual cameras in the 3-D space defining views of the model corresponding to the images; and

adding a new image to the sequence and processing the new image to obtain a camera solution for a corresponding new virtual camera for use in generating further model data;

a method of performing the processing of the new image comprising:

(a) identifying a plurality of image points in the new image which are matched to a respective plurality of image points of at least one preceding image of the sequence for which respective 3-D model data defining corresponding model points exists;

(b) determining a set of 2-D image co-ordinates of the identified image points in the new image and co-ordinates of respective model points; and

(c) processing the set of 2-D image point co-ordinates and respective 3-D model point co-ordinates to obtain the camera solution for the new image using a solving process in which the position and orientation of an image plane representative of the new virtual camera are calculated from a geometrical relationship in the 3-D model space between model points and image points defined by the set of co-ordinates.

28. A computer program comprising processor implementable instructions for carrying out a method as claimed in any one of claims 1 to 12 and 25.

29. A storage medium storing processor implementable instructions for carrying out a method as claimed in any one of claims 1 to 12 and 25.

30. An electrical signal carrying processor implementable instructions for carrying out a method as claimed in any one of claims 1 to 12 and 25.